Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

Lab Work

“Lab-2”

[Code No: COMP-342]

Submitted by:

Bibhushan Saakha [41]

Submitted to:

Mr Dhiraj Shrestha

Department of Computer Science and Engineering

Submission Date:

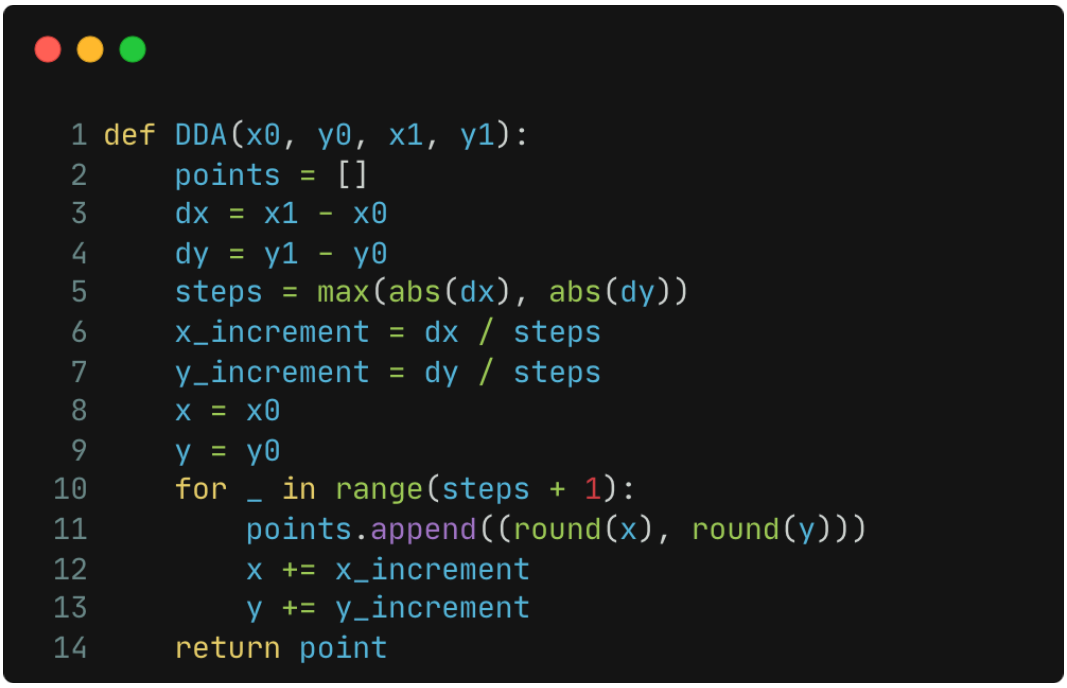
2024-05-19

Questions:

1. Implement Digital Differential Analyzer Line drawing algorithm.
2. Implement Bresenham Line Drawing algorithm for both slopes(|m|<1 and |m|>=1).
3. Implement the given line drawing algorithm to draw a line histogram for any given frequency inputs.

1. DDA Line Drawing Algorithm:

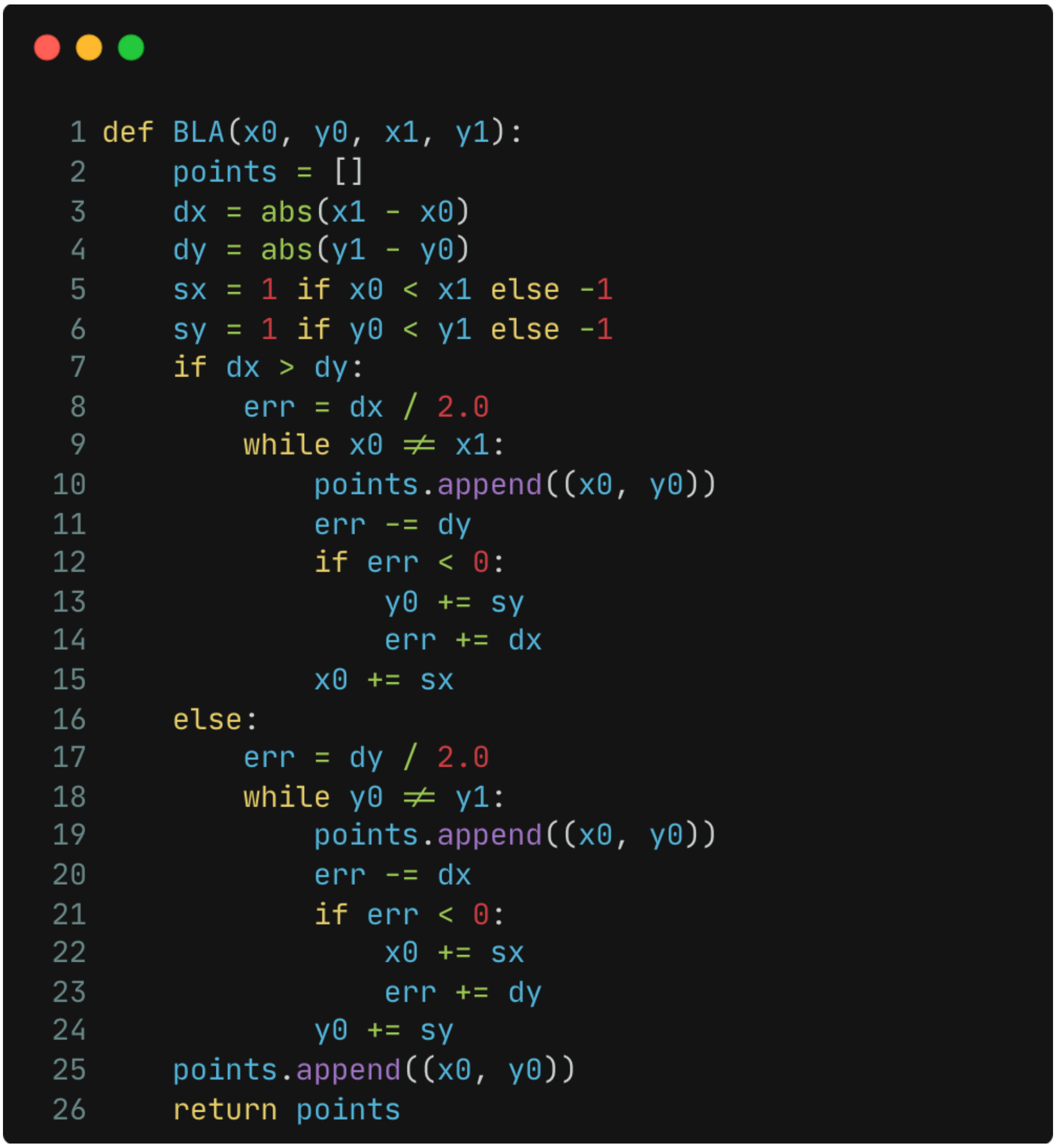
This function takes start and end points of the line and returns points to be used to plot the line using the Digital Differential Line Drawing Algorithm.



```python
def DDA(x0, y0, x1, y1):
    points = []
    dx = x1 - x0
    dy = y1 - y0
    steps = max(abs(dx), abs(dy))
    x_increment = dx / steps
    y_increment = dy / steps
    x = x0
    y = y0
    for _ in range(steps + 1):
        points.append((round(x), round(y)))
        x += x_increment
        y += y_increment
    return point
```

2. Bresenham Line Drawing Algorithm:

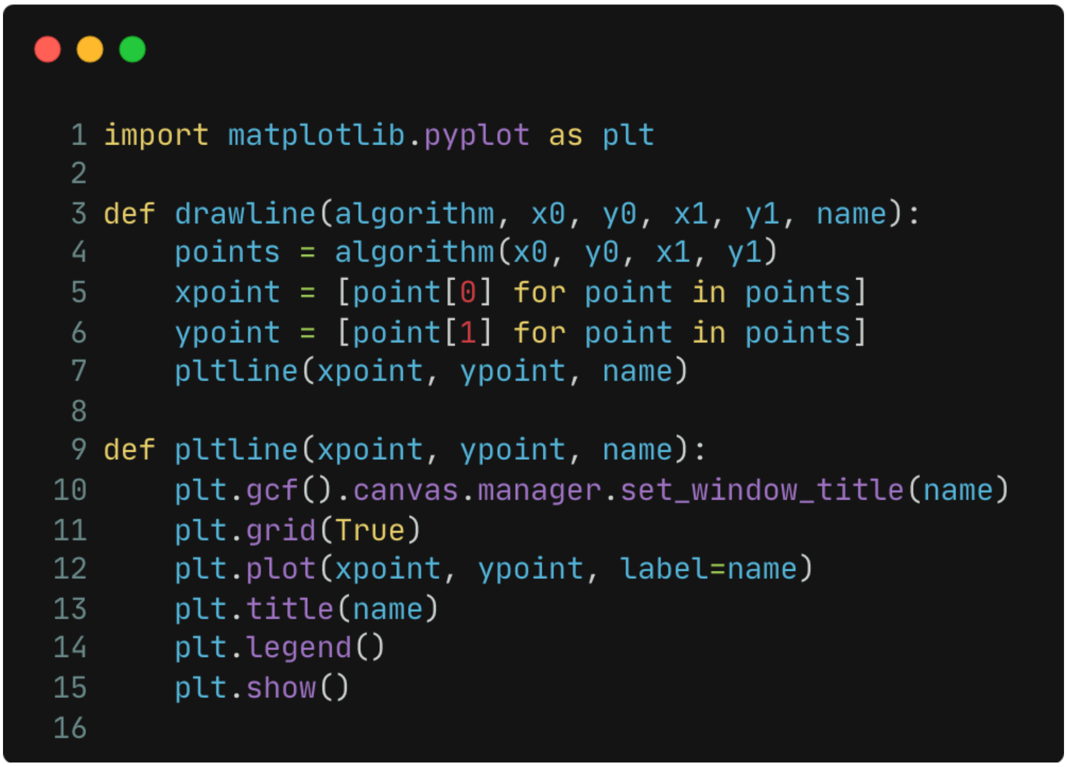
This function takes start and end points of the line and returns points to be used to plot the line using Bresenham Line Drawing Algorithm.



```python
def BLA(x0, y0, x1, y1):
    points = []
    dx = abs(x1 - x0)
    dy = abs(y1 - y0)
    sx = 1 if x0 < x1 else -1
    sy = 1 if y0 < y1 else -1
    if dx > dy:
        err = dx / 2.0
        while x0 ≠ x1:
            points.append((x0, y0))
            err -= dy
            if err < 0:
                y0 += sy
                err += dx
            x0 += sx
    else:
        err = dy / 2.0
        while y0 ≠ y1:
            points.append((x0, y0))
            err -= dx
            if err < 0:
                x0 += sx
                err += dy
            y0 += sy
    points.append((x0, y0))
    return points
```

3. Line Plotter

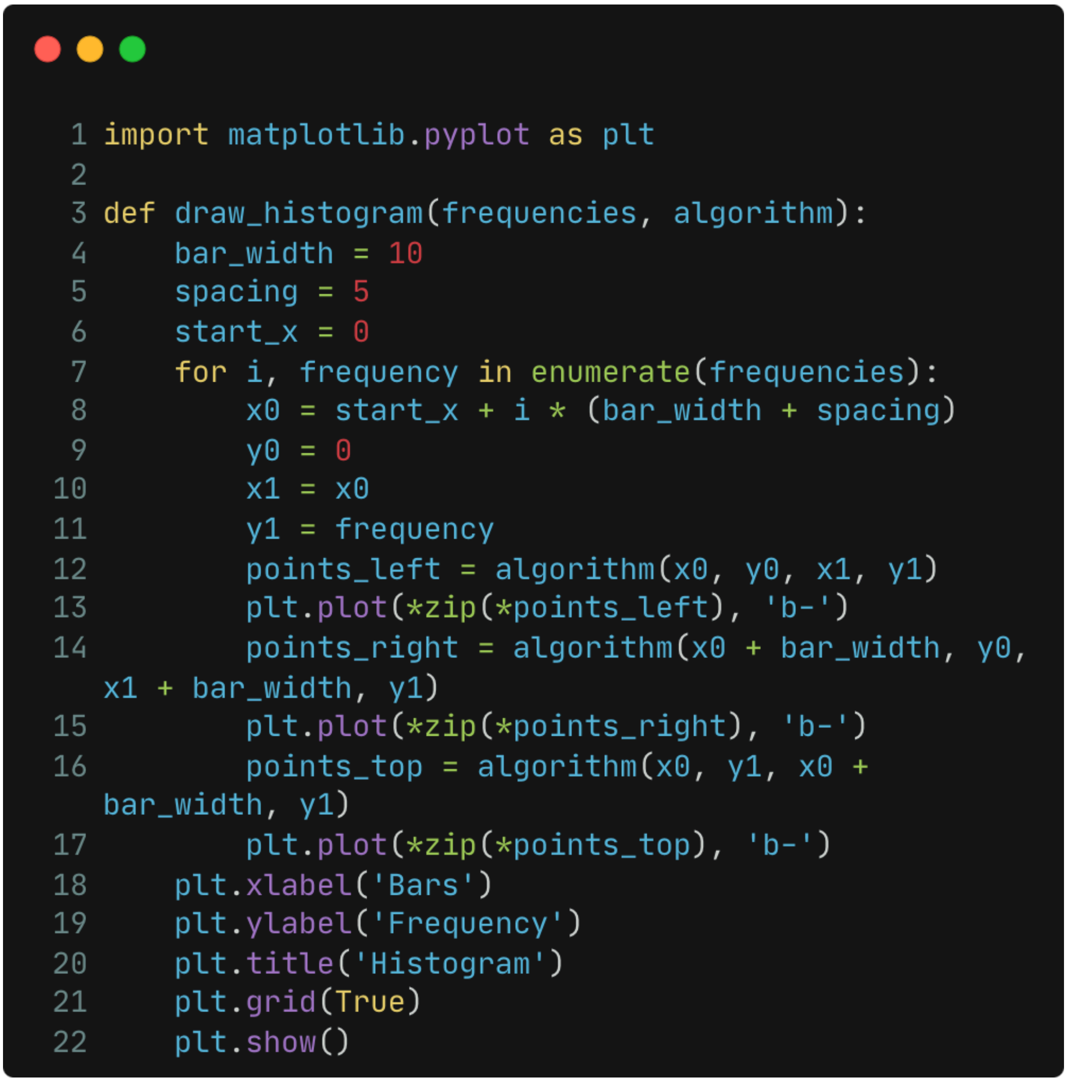
This function takes the endpoints and the algorithm to be used and calls the algorithm and uses the returned points to draw the line.



```python
import matplotlib.pyplot as plt

def drawline(algorithm, x0, y0, x1, y1, name):
    points = algorithm(x0, y0, x1, y1)
    xpoint = [point[0] for point in points]
    ypoint = [point[1] for point in points]
    pltline(xpoint, ypoint, name)

def pltline(xpoint, ypoint, name):
    plt.gcf().canvas.manager.set_window_title(name)
    plt.grid(True)
    plt.plot(xpoint, ypoint, label=name)
    plt.title(name)
    plt.legend()
    plt.show()

```

4. Histogram Generator

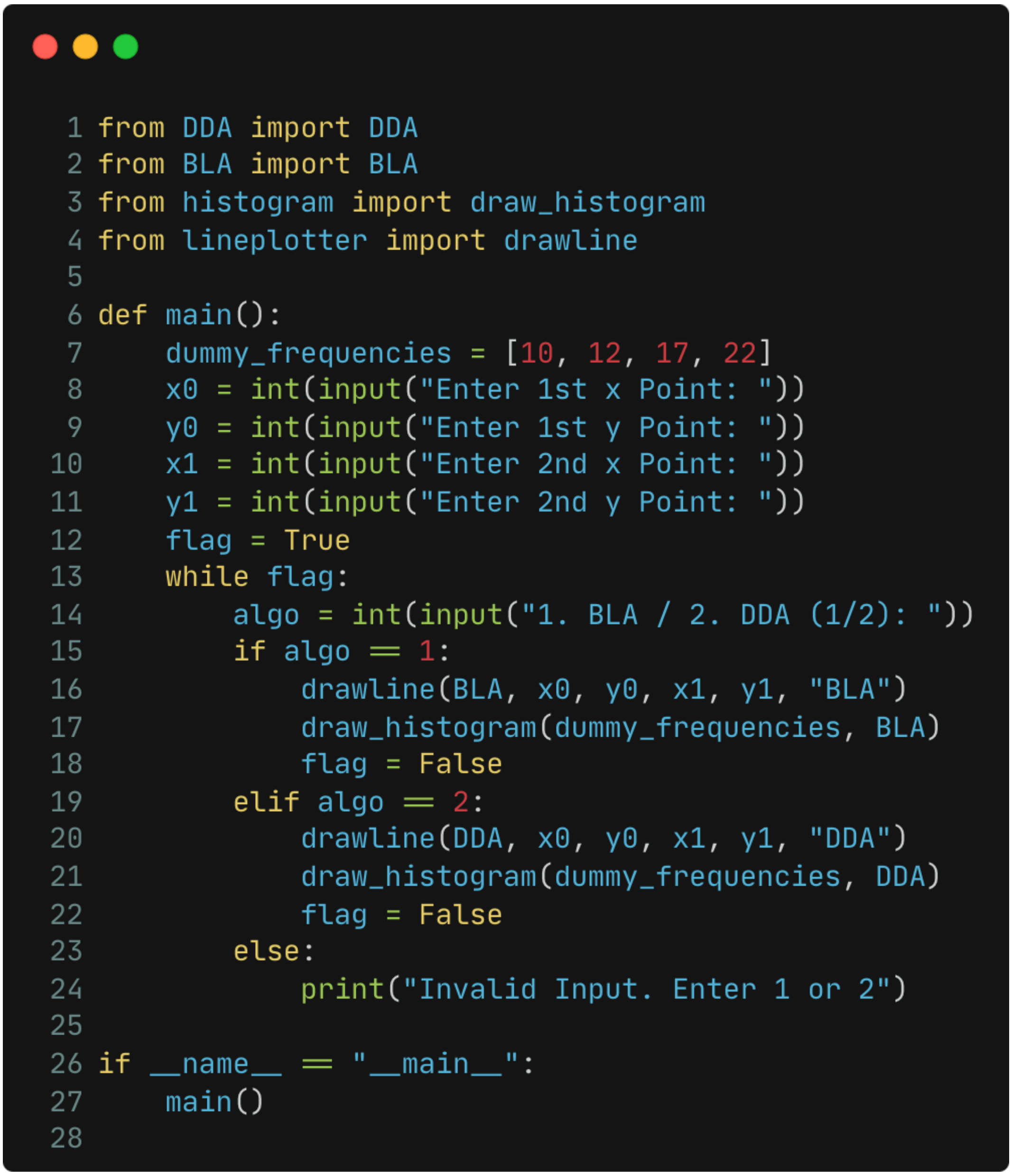
This function takes the frequencies and the algorithm to be used and then uses the algorithm to generate histogram of the given frequencies.



```python
import matplotlib.pyplot as plt

def draw_histogram(frequencies, algorithm):
    bar_width = 10
    spacing = 5
    start_x = 0
    for i, frequency in enumerate(frequencies):
        x0 = start_x + i * (bar_width + spacing)
        y0 = 0
        x1 = x0
        y1 = frequency
        points_left = algorithm(x0, y0, x1, y1)
        plt.plot(*zip(*points_left), 'b-')
        points_right = algorithm(x0 + bar_width, y0, x1 + bar_width, y1)
        plt.plot(*zip(*points_right), 'b-')
        points_top = algorithm(x0, y1, x0 + bar_width, y1)
        plt.plot(*zip(*points_top), 'b-')
    plt.xlabel('Bars')
    plt.ylabel('Frequency')
    plt.title('Histogram')
    plt.grid(True)
    plt.show()
```

5. Main.py

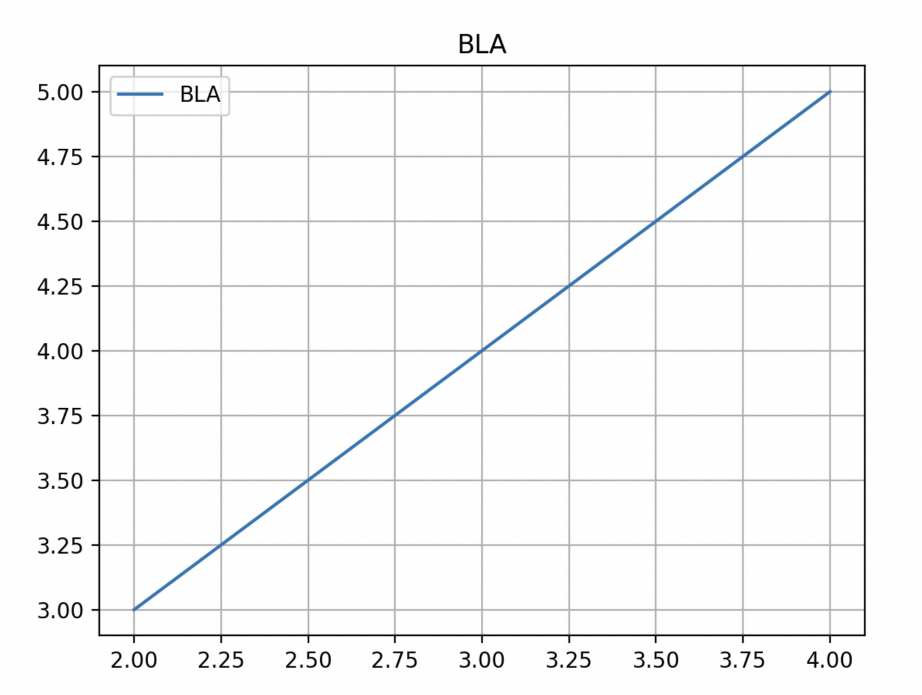
This uses all the above mentioned function to take start and end points from user and generate a line followed by a histogram using the algorithm of their choice.



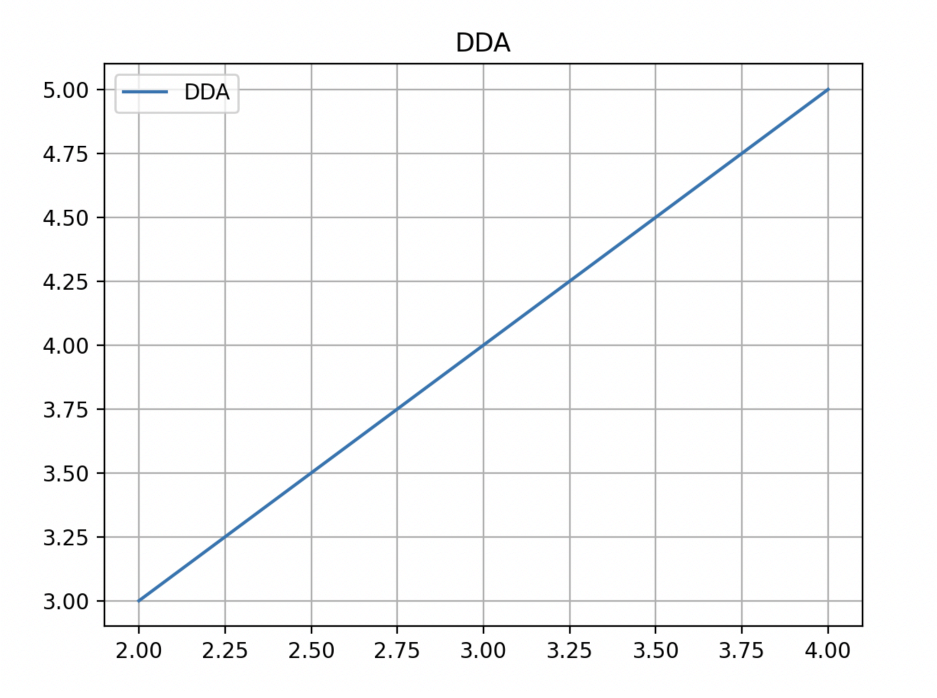
```python
from DDA import DDA
from BLA import BLA
from histogram import draw_histogram
from lineplotter import drawline

def main():
    dummy_frequencies = [10, 12, 17, 22]
    x0 = int(input("Enter 1st x Point: "))
    y0 = int(input("Enter 1st y Point: "))
    x1 = int(input("Enter 2nd x Point: "))
    y1 = int(input("Enter 2nd y Point: "))
    flag = True
    while flag:
        algo = int(input("1. BLA / 2. DDA (1/2): "))
        if algo == 1:
            drawline(BLA, x0, y0, x1, y1, "BLA")
            draw_histogram(dummy_frequencies, BLA)
            flag = False
        elif algo == 2:
            drawline(DDA, x0, y0, x1, y1, "DDA")
            draw_histogram(dummy_frequencies, DDA)
            flag = False
        else:
            print("Invalid Input. Enter 1 or 2")

if __name__ == "__main__":
    main()

```

Outputs:

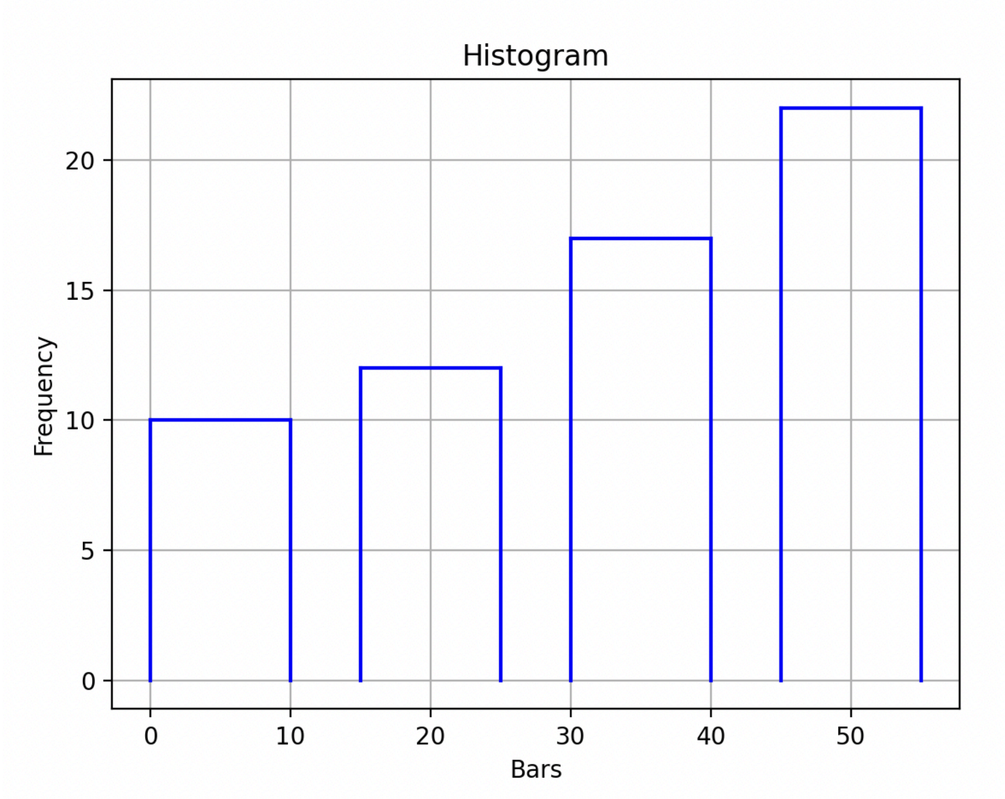
a. Line Using BLA:



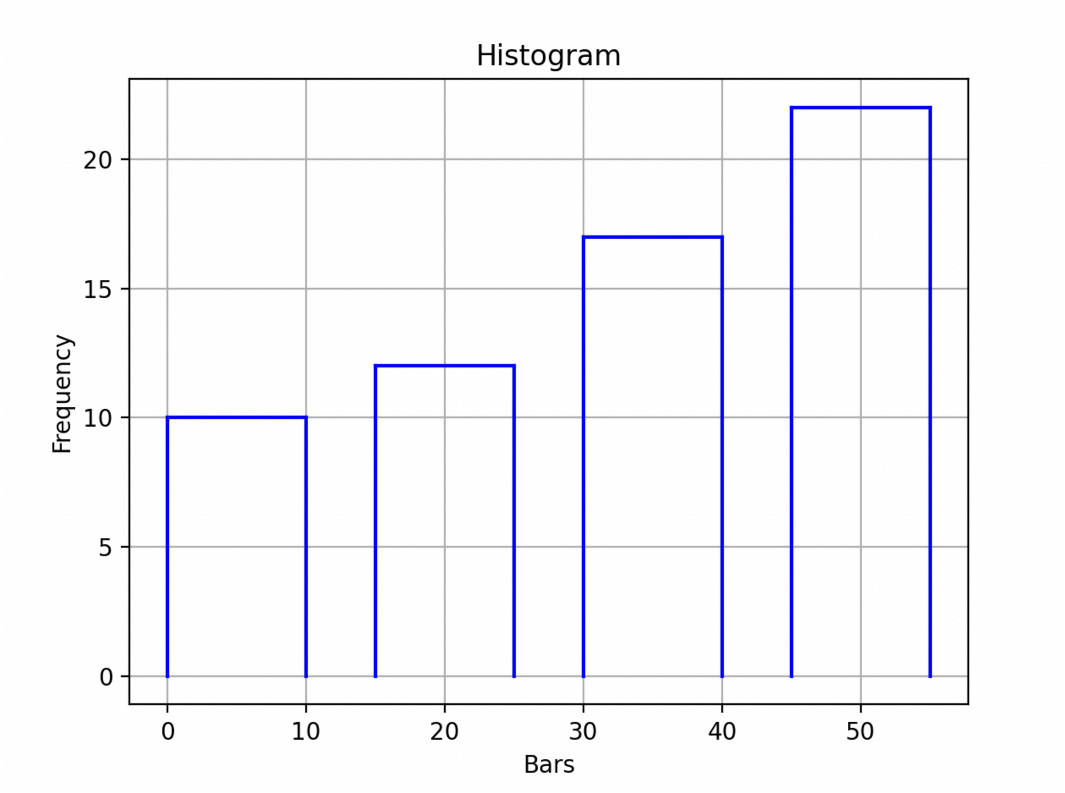
b. Line Using DDA:



b. Histogram Using BLA:



b. Histogram Using DDA:


Histogram
Frequency
Bars